

REMARKS

This Amendment is in response to the Office Action mailed April 18, 2003 and the Office Action mailed September 10, 2003. Claims 1-15 and 17-25, and 27-43 are now in the application. Claims 19 and 25 were previously amended in the Response filed July 16, 2003. Claims 1, 2, 9, 15, and 35 are currently amended.

Claim Rejections – 35 U.S.C. § 101

On July 1, 2003, applicants' attorney, Examiner Steelman and Supervisory Examiner Wei discussed the rejection of claims 19 and 25 under 35 USC 101. It was agreed that if the preamble of these claims was amended to reflect execution of instructions and accomplishment of a functional objective, then the rejection of claims 19-30 under 35 USC 101 would be overcome. Independent claims 19 and 25 were so amended on July 16, 2003.

Claim Rejections – 35 U.S.C. § 103

Claims 1, 2, 6-9, 11-13, 15-18, 31-35, and 39-41 were rejected under 35 U.S.C. § 103(a) as being anticipated by Breslau et al. (USPN 6,345,311), in view of Srivastava (USPN 5,966,539).

The Breslau reference shows a heterogeneous environment and does not show heterogeneous programs. Also, and most important, Breslau is moving homogeneous objects around the heterogeneous environment. In moving to a new environment an object is instantiated from source and compiled by the appropriate compiler for the new environment. There is no translation of components in a heterogeneous program having a plurality of components of different form. (See col. 5, line 66 to col. 6, line 13)

In the present invention, the programs are heterogeneous meaning the programs have a plurality of components whose instructions in the components differ in form from one component to another. Transforming components having instructions of differing form into intermediate representation is not the same as recompiling a homogeneous program from source code.

The Srivastava reference also shows only homogeneous programs. All modules of a program are linked together into a single module (See linker 50, FIG. 2) before being optimized

by optimizer 60 (FIG. 2) The optimizer is shown in detail in FIG. 3 of Srivastava reference. The modules can only be linked and optimized as a single large module if the instructions are of the same form. For this reason, the program in Srivastava reference must be homogeneous.

Srivastava reference does not operate on a heterogeneous program having a plurality of components differing in form, i.e. where the components are written to run on differing systems or architectures. Further, Srivastava links all modules into a single large module as this is the only way the reference can perform a compatible transformation for all modules of the program. Srivastava transforms the whole program and not each module, separately.

A definition of heterogeneous programs is stated in the present above-identified application as follows:

"In a new programming paradigm, a program is now a collection of components. Each component publishes an interface without exposing its inner details. Thus, a component can internally exist in any form: Intel x86 binary, Intel IA-64 binary, Visual Basic (VB) byte codes, Java class files, or any Virtual Machine (VM) binary. A heterogeneous program consists of components in different forms." (Page 2, lines 3-7).

Since the components of the heterogeneous programs are in different forms, each component must be transformed individually rather than operating on the program as a whole as is done in the Srivastava reference.

With these differences in mind, the manner in which claims 1, 2, 6-9, 11-13, 15-18, 31-35, and 39-41 differ from the prior art references will now be reviewed.

Claim 1 is directed to a method for translating and transforming a heterogeneous program having a plurality of components. Further, the method operates on each component to obtain a binary, to determine a plurality of basic blocks, translate platform-specific instructions into intermediate representations, and create intermediate representations of code blocks and procedures to create an intermediate representation of the component. As discussed above, neither the Breslau reference or the Srivastava reference operate on a heterogeneous program with a plurality of components. Further, Breslau moves objects to different environments and recompiles from source in each move. Srivastava operates on an entire homogeneous program (all modules linked into a single module) and does not operate on components of a

heterogeneous program as discussed above. Because of these differences, Claim 1 should be allowed.

Claims 2, 6-9, and 11-13 depend from claim 1 and should be allowed for the same reasons as just discussed for claim 1.

Claim 15 is directed to computer readable medium having instructions to perform a reader method that reads a heterogeneous program having a plurality of components in at least two different forms, obtains a binary for each component and builds an intermediate representation for each component and the program containing the components. Breslau does not operate on a heterogeneous program with a plurality of components of different form. Breslau only recompiles homogeneous objects from source. Srivastava does not operate on a heterogeneous program with a plurality of components of different form. Srivastava only operates on a homogeneous program as a whole. Claim 15 should be allowed.

Claim 18 depends from claim 15 and should be allowed for the same reasons as discussed above for claim 15.

Claim 31 is directed to a computing system having a translation and transformation system to translate a platform specific binary corresponding to a component in a heterogeneous program into a plurality of intermediate representation instructions. As discussed above neither Breslau reference or Srivastava reference translate a component of a heterogeneous program. Heterogeneous programs differ from homogeneous programs as discussed above. Claim 31 should be allowed.

Claims 32-34 depend from claim 31 and should be allowed for the same reasons as discussed above for claim 31.

Claim 35 is directed to a computer-readable medium storing instructions for a method for transforming a heterogeneous program having a plurality of components. The method translates platform-specific binary for each component in the heterogeneous program into intermediate representation instructions. Breslau reference recompiles homogeneous objects and does not operate on heterogeneous programs. Srivastava reference operates on a homogeneous program as a whole unit and does not translate each component in a heterogeneous program. Claim 35 should be allowed.

Claims 39-40 depend from claim 35 and should be allowed for the same reasons as discussed above for claim 35.

Claim 41 is directed to a computer-readable medium having instructions for a method that iterates an intermediate representation of a heterogeneous program to create a plurality of new versions. Neither the Breslau reference or the Srivastava reference teaches operating on a heterogeneous program as discussed above. Claim 41 should be allowed.

Claims 3-5, 10, 36-38, 42, and 43 were rejected under 103(a) as being unpatentable over Breslau et al., in view of Srivastava, as applied to claims 1, 6, 35 and 41, and further in view of Srivastava (USPN 5,539,907) i.e. Srivastava2 reference.

The Srivastava2 reference describes the same linker 50 and optimizer 60 in FIGs. 2 and 3 as in the first cited Srivastava reference. As such it does not make up for any of the deficiencies of the first Srivastava reference discussed above for independent claims 1, 35 and 41. Srivastava2 reference is still directed to operating on a homogeneous program as a whole and does not operate on components of a heterogeneous program.

Claims 3-5 and 10 depend from claim 1 and should be allowed for the same reasons as discussed above for claim 1.

Claims 36-38 depend from claim 35 and should be allowed for the same reasons as discussed above for claim 35.

Claims 42-43 depend from claim 41 and should be allowed for the same reasons as discussed above for claim 41.

Claim 14 was rejected under 103(a) as being unpatentable over Breslau, in view of Srivastava '539, as applied to claim 13, and further in view of common knowledge in computer art.

Claim 13 depends from claim 1 and should be allowed for the same reasons as discussed above for claim 1.

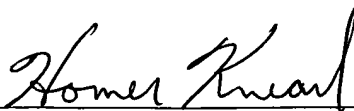
As all claims now in the application are in condition for allowance, Applicants request the application be allowed and pass to issuance as soon as possible.

It is believed that no further fees are due with this Response. However, the commissioner is hereby authorized to charge any deficiencies or credit any overpayment with respect to this patent application to deposit account number 13-2725.

Respectfully submitted,

Dated: October 09, 2003





Homer L. Knearl, Reg. No. 21,197
MERCHANT & GOULD P.C.
P. O. Box 2903
Minneapolis, MN 55402-0903
(303) 357-1633